

A Comparison of On-Demand and Table Driven Routing for Ad-Hoc Wireless Networks

Jyoti Raju J.J. Garcia-Luna-Aceves
Computer Science Dept. Computer Engineering Dept.
Baskin School of Engineering
University of California, Santa Cruz
jyoti, jj@cse.ucsc.edu

Abstract—We introduce WRP-Lite, which is a table-driven routing protocol that uses non-optimal routes, and compare its performance with the performance of the dynamic source routing (DSR) protocol, which is an on-demand routing protocol for wireless ad-hoc networks. We evaluate the performance of WRP-Lite and DSR for varying degree of mobility and traffic in a 20-node network. The performance parameters are end-to-end delay, control overhead, percentage of packets delivered, and hop distribution. We show that WRP-Lite has much better delay and hop performance while having comparable overhead to DSR.

I. INTRODUCTION

Ad-hoc networks (or multi-hop packet-radio networks) consist of mobile hosts interconnected by routers that can also move. Considerable work has been done in the development of routing protocols for ad-hoc networks, starting from the seventies with work on the DARPA PRNET and SURAN projects [5], [17], [12]. In recent years, the interest in ad-hoc networks has grown due to the availability of wireless communication devices that work in the ISM bands in the U.S.

Routing for ad-hoc networks can be classified in two main types: *table-driven* and *on-demand*. Table driven routing attempts to maintain consistent information about the path from each node to every other node in the network. The Destination-Sequenced Distance-Vector Routing (DSDV) protocol is a table driven algorithm that modifies the Bellman-Ford routing algorithm to include timestamps that prevent loop-formation [15]. The Wireless Routing Protocol (WRP) is a distance vector routing protocol which belongs to the class of path-finding algorithms that exchange second-to-last hop to destinations in addition to distances to destinations [13]. This extra information helps remove the “counting-to-infinity” problem that most distance vector routing algorithms suffer from. It also speeds up route convergence when a link failure occurs.

On-demand routing protocols were designed with the aim of reducing control overhead, thus increasing bandwidth and conserving power at the mobile stations. These protocols limit the amount of bandwidth consumed by maintaining routes to only those destinations for which a source has data traffic. Therefore, the routing is source-initiated as opposed to table-driven routing protocols that are destination initiated. There are several recent examples of this approach (e.g., AODV [16], ABR [20], DSR [10], TORA [14], SSA [4], ZRP [9]) and the routing protocols differ on the specific mechanisms used to disseminate flood-search packets and their responses, cache the information heard from other nodes’ searches, determine the cost of a link, and determine the existence of a neighbor. However, all the on-demand routing proposals use flood search messages that either: (a) give sources the entire paths to destinations, which are then used in source-routed data packets (e.g., DSR); or (b) provide only the distances and next hops to destinations, validating them with sequence numbers (e.g., AODV) or time stamps (e.g., TORA).

Several studies have been published comparing the performance of the above routing protocols using different simulators, mobility models and performance metrics. One of the first comprehensive studies was done by the Monarch project of CMU, the results of which are presented in [2]. This study compared DSDV, AODV, DSR and TORA and

introduced some standard metrics that may be used in further studies of wireless routing protocols. A paper by Das et al. [6] compares a larger number of protocols. However, link level details and MAC interference are not modelled. This may not give an adequate reflection of the delays suffered by packets that are made to wait while the MAC protocol acquires the channel. It also does not reflect how high data traffic rate may interfere with routing protocol convergence. Another recent study [3] compares the same protocols as the work by Broch, et al.[2]. This study used specific scenarios to test the protocol behavior. Based on their results, all of these papers conclude that on-demand routing protocols perform better than table-driven routing protocols. However, all the table-driven routing protocols tested use the optimum routing approach. In other words, these protocols try to maintain shortest paths at all times. A consequence of maintaining shortest paths is that if the topology of the network changes rapidly, the control overhead increases dramatically. In this paper, we show that relaxing the requirement for shortest paths in a table driven routing protocol can lead to solutions whose performance is equivalent or even better than the performance of on-demand routing approaches. Our goal is to design a table-driven distance-vector routing protocol that uses the same constraints used in on-demand routing protocols, i.e. paths are used as long as they are valid and updates are only sent when a path becomes invalid. To this end, we adapt WRP, to provide non-optimum routing and we call the resulting protocol WRP-Lite. The reason why prior table-driven routing protocols have been unable to perform non-optimum routing is that these protocols have used either distances to destinations or topology maps to predict paths to destinations. None of these techniques allow a router to discern if the path picked by it conflicts with its neighbors, resulting in “counting to infinity” problems. Consequently, these protocols have to send updates in order to avoid loops, and the best that can be done is that the updates are sent periodically. However, in WRP-Lite, the paths used by neighbors are maintained and this allows the design of a distance-vector protocol with non-optimum routing and event-driven updates, resulting in reduced control overhead.

Section II presents a brief description of WRP-Lite, and illustrates the key aspects of difference between WRP and WRP-Lite. Section III, presents simulations that compare the performance of WRP-Lite and DSR in random mobility scenarios. We chose DSR, because DSR has been shown to outperform other on-demand routing algorithms in previous studies [2], [3]. Finally, Section IV, presents our conclusions.

II. PROTOCOL DESCRIPTION

A. Network Model

A network is modelled as an undirected graph $G(V, E)$ which can have partitions. V is the number of nodes in the network and E is the number of links in the network. A node principally consists of a router, which may be physically attached to multiple IP hosts (or IP-addressable devices). Instead of having interface identifiers, a router has a single node identifier, which helps the routing and other application protocols identify it. In a wireless network, a node has radio connectivity with multiple nodes using a single physical radio link. Accordingly, we map a physical broadcast link connecting a node and its multiple neighbors into point-to-point links between the node and its neighbors. Each link has a positive cost associated with it. If a link

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 2000		2. REPORT TYPE		3. DATES COVERED 00-00-2000 to 00-00-2000	
4. TITLE AND SUBTITLE A Comparison of On-Demand and Table Driven Routing for Ad-Hoc Wireless Networks				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of California at Santa Cruz, Department of Computer Engineering, Santa Cruz, CA, 95064				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 5	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

fails, its cost is set to infinity. A node failure is modelled as all links incident on the node getting set to infinity.

For the purpose of routing table updating, a node A considers a node B as its neighbor if it hears update messages from node B . Node B is no longer node A 's neighbor when node A cannot send data packets to it.

Routing messages are broadcast unreliably and the protocol assumes that routing packets may be lost due to changes in link connectivity, fading or jamming. A neighbor protocol is used that brings up a link when it hears sufficient number of packets from a neighbor. The link is brought down when a unicast data packet can no longer be sent to the neighbor despite retransmissions at the link layer. The functionality of such a neighbor protocol can be easily added onto a MAC protocol like (e.g., IEEE802.11), TDMA, or any of the various dynamic scheduling MAC protocols proposed recently [18], [11] without requiring additional network-level control packets.

B. Routing Structures maintained

The routing structures maintained in WRP-Lite are a subset of those maintained by WRP, i.e., a *routing table* and a *distance table*. Since messages are assumed to be transmitted unreliably, no *message retransmission list* is required. WRP-Lite also does not maintain any packet buffer for data packets waiting for routes. Packets are sent if there is a valid route and they are dropped if there is no valid path at the moment of arrival.

The routing table at router i contains entries for all known destinations. Each entry consists of the destination identifier j , the successor to that destination s_j^i , the second-to-last-hop to the destination p_j^i , the distance to the destination D_j^i and a route tag tag_j^i . When the element tag_j^i is set to *correct*, it implies a loop-free finite value route. When it is set to *null*, it implies that the route still has to be checked and when it is set to *error*, an infinite metric route or a route with a loop is implied.

The distance table at router i is a matrix containing, for each known neighbor k and each destination j , the distance value of the route from i to j through k , D_{jk}^i and the second-to-last hop p_{jk}^i on that route. D_{jk}^i is always set equal to $RD_{jk}^k + l_{ik}^i$, where RD_{jk}^k is the distance reported by k to j in the last routing message and l_{ik}^i is the link cost of link (i, k) . The link cost may be set to one reflecting hop count or it may be set to some other link parameter like latency, bandwidth, etc.

C. Routing information exchanged

Routing update messages are broadcast to all neighbors. Each packet contains the address of the sender and a list of routing table entries, where each entry specifies a destination, the distance to the destination and the predecessor to the destination. If the MAC layer allowed for transmission of reliable updates with no retransmission overhead (e.g., [19]), only incremental routing updates need to be sent. In this paper, however, we assume a MAC protocol based on collision avoidance. To avoid collisions of data packets with other packets caused by hidden terminals, such protocols require nodes to defer for fixed periods of time after detecting carrier [8]. Accordingly, sending larger control packets does not decrease throughput at the MAC layer, because the overhead ($RTS - CTS$ exchange) for the MAC protocol to acquire the channel does not depend on packet size. Therefore, in the rest of the paper, we assume that routers transmit their entire routing tables when they send control messages. Control packet size may affect the delay experienced by data packets in the MAC layer. However, as our simulations show, this does not happen because the number of control packets we generate is substantially low.

All data packets contain the source and the destination and are unicast reliably by the link layer.

D. Routing-Table Updating

Routing tables are updated under two conditions, the first condition being the receipt of an update message and the second condition being a detection of a link status change.

D.1 Receiving an update

The processing of an update in WRP-Lite is done in the same manner as in WRP. When an update from neighbor k is received, the entries in the distance table corresponding to neighbor k are updated. The paths to each destination are then recomputed. WRP-Lite sends updates only if any of the following conditions have been met.

1. A node discovers a new destination with a finite and valid path to the destination.
2. A node loses the last path to a destination.
3. A node suffers a distance increase to a destination.

From the above conditions, it follows that an update is not sent if a next hop to destination changes. It is also not sent if the distance to a destination decreases. However, an update is sent when the distance to a destination increases, because this condition has the potential to cause a loop.

Two more conditions are added to prevent permanent looping due to unreliable broadcasts.

4. A node sends a unicast update to a neighbor that sends it a data packet, if the neighbor is upstream from it towards the destination.
5. A node sends a unicast update to a neighbor that sends it a data packet, when the path implied by the neighbor's distance table entry is different from the path implied by the node's routing table.

In both these conditions, the data packets are dropped. Permanent looping can occur when nodes are not aware of the latest changes in their neighbor's routing tables. The use of conditions 4 and 5 can be explained with the help of an example shown in Fig. 1.a. The node addresses are marked in bold font. Node j is the required destination. The path to j implied by traversing predecessors from j is marked in italics. Initially, all nodes have loop-free routes. The loss of links (i, j) and (m, j) and the loss of update packets from i and m can result in a loop shown in Fig. 1.b. When i gets a data packet from k , it finds that its distance table entry for k implies the path ij , while i 's own path implies $ilmj$ which is different from ij . Due to condition 5, the data packet is dropped and a unicast routing update is sent resulting in k setting its path to kmj . Now, when k gets a data packet from m , it sends a unicast update to m because m is its successor on the path to j . This follows from condition 4. When m gets the update, it detects a loop and resets its distance to infinity, thus breaking the loop.

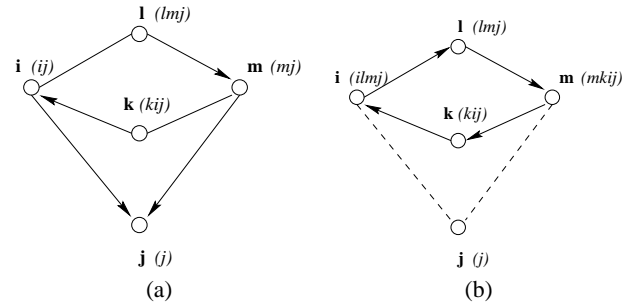


Fig. 1. Creation of a permanent loop due to unreliable updates

D.2 Topology and Link-Cost Changes

When a MAC protocol can no longer send a data packet to a neighbor, the link to the neighbor is marked with value infinity, and all the distances are recomputed. If the path to any destination is lost, then an update is sent.

When the routing protocol gets a link up signal from the neighbor protocol, it broadcasts an update and includes the neighbor k in the distance table with all distances through k set to infinity. One exception is the distance from i to k through k , which is set to one.

III. PERFORMANCE EVALUATION

We ran a number of simulation experiments to compare the performance of WRP-Lite against DSR. These simulations allowed us to independently change input parameters and check the protocol's sensitivity to these parameters. Both the protocols are implemented in *CPT*, which is a C++ based toolkit that provides a wireless protocol stack and extensive features for accurately simulating the physical aspects of a wireless multi-hop network*. The protocol stack in the simulator can be transferred with a minimal amount of changes to a real embedded wireless router. The stack uses IP as the network protocol. The routing protocols directly use UDP to transfer packets. The link layer implements a medium access protocol very similar to the IEEE 802.11 standard [1] and the physical layer is based on a direct sequence spread spectrum radio with a link bandwidth of 1 Mbit/sec.

To run DSR in CPT, we ported the DSR code available in the *ns2* [7] wireless release. There are two differences in our DSR implementation as compared to the implementation used in [2]. Firstly, we do not use the *promiscuous* listening mode in DSR. Besides introducing security problems, this feature cannot be supported in any IP stack where the routing protocol is in the application layer and the MAC protocol uses multiple channels to transmit data. Secondly, the routing protocol in our stack does not have access to the MAC and link queues, which is the reason why we cannot reschedule packets that have already been scheduled over a link for DSR. Table I shows the constants used in the implementation of DSR.

TABLE I
CONSTANTS USED IN DSR SIMULATION

Time between ROUTE REQUESTS (exponentially backed off)	500(msecs)
Size of source route header carrying carrying n addresses	$4n+4$ (bytes)
Timeout for Ring 0 search	30(msecs)
Time to hold packets awaiting routes	30 (secs)
Max number of pending packets	50

A. Scenarios used in comparison

We compared the protocols using two traffic scenarios. In both scenarios, we used the "random waypoint" model described in [2]. In this model, each node begins the simulation by remaining stationary for *pause time* seconds and then selecting a random destination and moving to that destination at a speed of 20 m/s. Upon reaching the destination, the node pauses again for *pause time* seconds, selects another destination, and proceeds there as previously described, repeating this behavior for the duration of the simulation. We used the speed of 20m/s, which is approximately the speed of a vehicle, because it has been used in simulations in earlier papers [2], [3] and thus provides a basis for comparison with other protocols. Two nodes can hear each other if the attenuation value of the link between them is such that packets can be exchanged with a probability p , where $p > 0$. The attenuation value between two nodes 1 and 2 is calculated using the following equation,

$$156 + 40\log(d) - 15\log(h1) - 15\log(h2) - g1 - g2 \quad (1)$$

where d is the distance in miles, $h1$ is the height of antenna 1 in feet, $h2$ is the height of antenna 2 in feet (both set to 20) and $g1$ is the gain

of antenna 1 and $g2$ is the gain of antenna 2 (both set to 3). Thus, at a distance of 1 mile, the attenuation is 111 db. Attenuation values are recalculated every time a node moves.

In both scenarios, we used a 20 node ad-hoc network, moving over a flat space of dimensions 4.2 X 3.1 miles (6.6 X 4.8 km) and initially randomly distributed with a density of approximately one node per square mile. We have random data flows, where each flow is a peer-to-peer constant bit rate (CBR) flow with an interarrival time of 250 msec between data packets. The data packet size is kept constant at 64 bytes. Data flows were started at times uniformly distributed between 20 and 120 seconds and they go on till the end of the simulation. The pause times are varied: 0, 30, 60, 120, 300, 600 and 900 seconds as done in [2].

In the first scenario, there are eight CBR sources, each of which establishes a connection with a randomly picked destination. All of the destinations are different from each other. The results are averaged over the three runs of the simulation with randomly generated source-destination pairs.

In the second scenario, we use 16 CBR sources. Since we model interference, our intention here is to see how the protocols perform as the cross traffic increases. Given that the overhead of table-driven routing protocols is independent of traffic, this scenario will also reflect on the scalability of the on-demand protocols. The results here are also averaged over three different runs with 16 distinct destinations in each run.

B. Metrics used

In comparing the two protocols, we use the following metrics:

- *Packet delivery ratio*: The ratio between the number of packets received by an application and the number of packets sent by the corresponding peer application.
- *Control Packet Overhead*: The total number of routing packets sent out during the simulation. Each broadcast packet is counted as a single packet.
- *Hop Count*: The number of hops a data packet took from the sender to the receiver.
- *End to End Delay*: The delay a packet suffers from leaving the sender application to arriving at the receiver application. Since dropped packets are not considered, this metric should be evaluated in conjunction with the metric of packet delivery ratio.

Packet delivery ratio gives us an idea about the effect of routing policy on the throughput that a network can support. It also is a reflection of the correctness of a protocol.

Control packet overhead has an effect on the congestion seen in the network and also helps evaluate the efficiency of a protocol. Low control packet overhead is desirable in low-bandwidth environments and environments where battery power is an issue.

In ad-hoc networks, it is sometimes desirable to reduce the transmitting power to prevent collisions. This will result in packets taking more number of hops to reach destinations. However, if the power is kept constant, the distribution of the number of hops data packets travel through is a good measure of routing protocol efficiency.

Delay has an effect on the throughput seen by reliable transport protocols like TCP. Average end-to-end delay is not an adequate reflection of the delays suffered by data packets. A few data packets with high delays may skew results. Therefore, we plot the cumulative distribution function of the delays. This plot gives us a clear understanding of the delays suffered by the bulk of the data packets.

C. Simulation results

C.1 Scenario 1: 8 sources

Fig. 2.a depicts the results for control packet overhead. The behavior of the protocols is very similar with WRP-Lite performing relatively better at higher rates of movement and plateauing off at lower speeds,

* We thank NOKIA wireless routers for making CPT available for our study.

while DSR performs better only for the case of no movement (pause time 900).

In Fig 2.b, we see that the percentage of data packets received are comparable for all both protocols, with DSR having a 2% edge over WRP-Lite.

For the next two graphs, the results are shown only for the highest mobility rate (pause time 0). Fig. 3.a shows the results of the distribution of hops taken by the data packets. This graphs depicts the noticeable difference between the routes taken by packets in an on-demand versus a table-driven protocol. Since WRP-Lite reacts to new links coming up, we notice that most packets take optimum paths. In fact, 50% of the packets take more optimal routes with WRP-Lite.

The most dramatic differences are seen in the delay performance shown in Fig. 3.b, which shows the delay in seconds on a logarithmic X axis. WRP-Lite has much better delay performance than DSR. Besides taking longer paths, packets also get delayed because they wait in buffers while routes are being searched for.

C.2 Scenario 2: 16 sources

This scenario of 16 sources was simulated with the purpose of evaluating the behavior of the protocols as the number of traffic sources increases. We typically expect an on-demand protocol to suffer as the number of traffic sources increase. As stated earlier, the graphs are averages over three runs to prevent topology specific skewing of results.

Fig. 4.a shows the results for control packet overhead. We see that DSR has an order of magnitude higher control overhead than WRP-Lite. As expected, the control overhead of WRP-Lite does not increase substantially due to increase in traffic.

Fig. 4.b depicts the results for the percentage of packets received. The performance of DSR suffer at pause time 0, with only 47% of the packets getting through to destinations, while WRP-Lite propagates 60% of the packets. For other pause times, the performance is very similar.

Figs. 5.a and 5.b both show results for the pause time 0. In Fig. 5.a, we see the hop distribution for the protocols, with WRP-Lite picking the most optimal paths. The delay distribution in Fig. 5.b shows similar results. Around 95% of the data packets are delivered within one second by WRP-Lite, while DSR delivers only 70% of the data packets within one second.

IV. CONCLUSIONS

Our work is a first step towards a more comprehensive evaluation of the comparative performance of on-demand and non-optimal table-driven routing algorithms. We introduced a version of WRP that provides non-optimum routes and used this as an example that table-driven routing can be just as efficient as on-demand routing in ad-hoc networks. WRP-Lite behaved better than DSR, an efficient on-demand routing protocol, in terms of hop distribution and delay, irrespective of the amount of traffic; furthermore, WRP-Lite also had lesser control overhead than DSR. This is because the protocol tolerates a certain degree of non-optimality in routes by using path information available at the routers.

A general observation that can be made is that in wireless networks protocol performance is linked very closely to the type of MAC protocol used. For instance, in DSR if the MAC protocol sends packets in bursts, we observe a lot more route error packets being sent in response to bursts of packet traveling on invalid paths. In conclusion, the design of the routing protocol should take into account the features of the lower layers in the wireless stack.

REFERENCES

[1] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. The Institute of Electrical and Electronics Engineers, 1997. IEEE Std 802.11.

[2] J. Broch et al. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proc. ACM MOBIKOM '98*, Dallas, TX, October 1998.

[3] P. Johansson et al. Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-Hoc Networks. In *Proc. IEEE/ACM Mobicom '99*, pages 195–206, Seattle, WA, Aug. 1999.

[4] R. Dube et al. Signal Stability-Based Adaptive Routing (SSA) for Ad-Hoc Mobile Networks. *IEEE Pers. Commun.*, February 1997.

[5] R. E. Kahn et al. Advances in Packet Radio Technology. *Proc. IEEE*, 66(11):1468–1496, Nov 1978.

[6] S. R. Das et al. Comparative Performance Evaluation of Routing Protocols for Mobile Ad-Hoc Networks. In *7th Int. Conf. on Comp. Communication and Networks*, pages 153–161, Lafayette, LA, Oct. 1998.

[7] Kevin Fall and Kannan Varadhan. *ns notes and documentation*. The VINT Project, UC Berkeley, LBL, USC/ISI and Xerox PARC, 1999. Available from <http://www-mash.cs.berkeley.edu>.

[8] C. L. Fullmer and J. J. Garcia-Luna-Aceves. Solutions to Hidden Terminal Problems in Wireless Networks. In *Proc. ACM SIGCOMM '97*, Cannes, France, September 1997.

[9] Z. Haas and M. Pearlman. The Performance of Query Control Schemes for the Zone Routing Protocol. In *Proc. ACM SIGCOMM '98*, Vancouver, British Columbia, August 1998.

[10] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad-Hoc Wireless Networks. *Mobile Computing*, 1994.

[11] J. Ju and V.O.K. Li. An Optimal Topology-Transparent Method in Multihop Packet Radio Networks. *IEEE/ACM Trans. Networking*, 6(3), June 1998.

[12] J. Jubin and J. Tornow. The DARPA Packet Radio Network Protocols. *Proc. IEEE*, 75(1):21–32, 1987.

[13] S. Murthy and J. J. Garcia-Luna-Aceves. An Efficient Routing Protocol for Wireless Networks. *ACM Mobile Networks and App. J.*, pages 183–197, Oct. 1996.

[14] V. D. Park and M. S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proc. IEEE INFOCOM '97*, Kobe, Japan, April 1997.

[15] C. E. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *Comp. Commun. Rev.*, pages 234–244, Oct 1994.

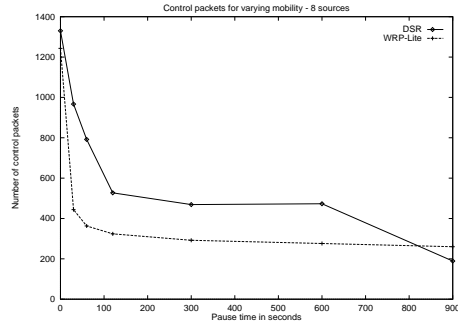
[16] C. E. Perkins and E. M. Royer. Ad-Hoc On-Demand Distance Vector Routing. In *Proc. 2nd IEEE Wksp. Mobile Comp. Sys. and Apps.*, pages 90–100, Feb 1999.

[17] N. Shacham and J. Westcott. Future Directions in Packet Radio Architectures and Protocols. *Proc. IEEE*, 75(1):83–98, Jan 1987.

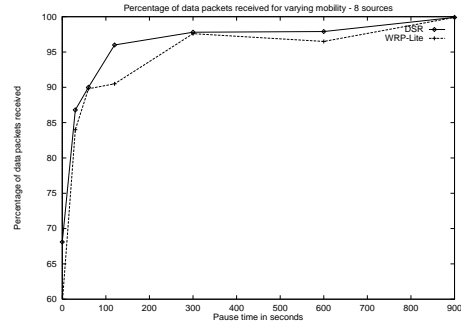
[18] T. Shepard. A Channel Access Scheme for Large Dense Packet Radio Networks. In *Proc. ACM SIGCOMM*, Aug 96.

[19] Z. Tang and J. J. Garcia-Luna-Aceves. A Protocol for Topology-Dependent Transmission Scheduling in Wireless Networks. In *Proc. of WCNC '99*, New Orleans, Louisiana, September 1999.

[20] C-K. Toh. A Novel Distributed Routing Protocol to support Ad-Hoc Mobile Computing. In *Proc. 1996 IEEE 15th Annual Int'l Phoenix Conf. Comp. and Commun.*, pages 480–86, Mar. 1996.

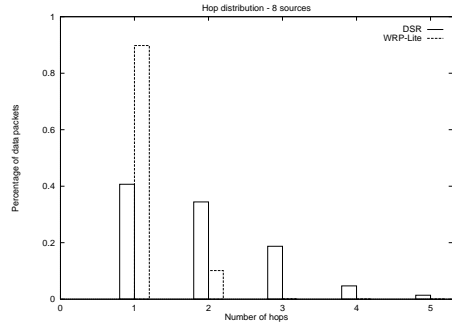


(a)
Number of control packets sent

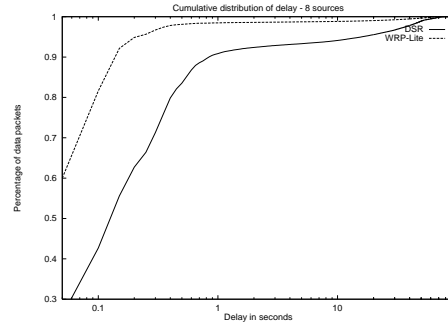


(b)
Percentage of data packets received

Fig. 2. 8 sources picking random destinations for peer-to-peer flow

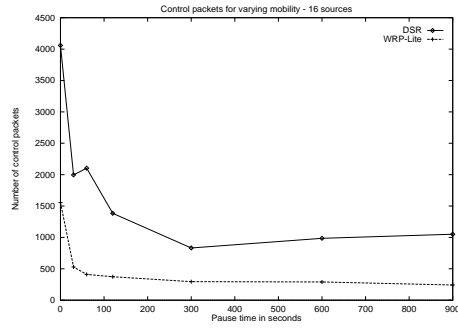


(a)
Hop distribution

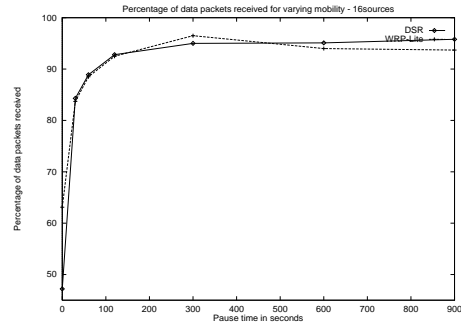


(b)
Cumulative delay distribution

Fig. 3. 8 sources picking random destinations for peer-to-peer flow for pause time 0

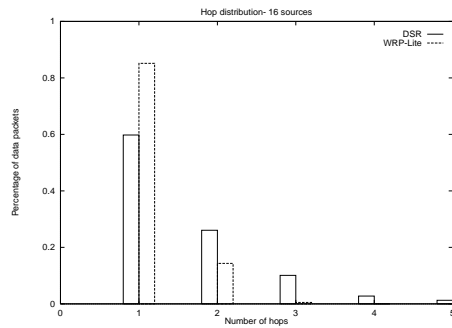


(a)
Number of control packets sent

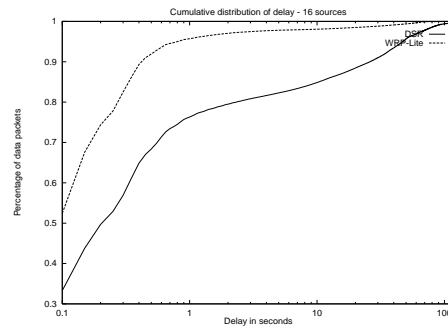


(b)
Percentage of data packets received

Fig. 4. 16 sources picking random destinations for peer-to-peer flow



(a)
Hop distribution



(b)
Cumulative delay distribution

Fig. 5. 16 sources picking random destinations for peer-to-peer flow for pause time 0